60 ideas from “Memory as a Computational Resource” (2020) by Samuel Gershman and Ishita Dasgupta

Link to actual paper

1. The recursive structure of a problem can be used to implement a space-time tradeoff
2. Redundant computation, which costs time, can be avoided by storing partial solutions in memory, which costs space
3. First, a general overview of computational reuse strategies
4. Then, examine 4 domains of cognitive science in which these strategies have been studied: mental arithmetic, mental imagery, probabilistic inference, planning
5. Computer scientists Donald Michie proposed memoization as a technique to reduce redundant computation
6. Function calls are intercepted by a memoizer that inspects a cache (the memo table) of past function calls and their outputs
7. If a function has previously been called with the same inputs, the previously computed result is reused.
8. If the target function itself has not been memoized, any sub functions that have been memoized can be intercepted as the target function is executed
9. When the input space is vast and there’s a low probability of a repeated function call, a memo table might be “too rigid” for some problems…
10. The generalization issue… the solution is to interpolate between cached outputs.
11. Memoization must be selective about what it stores and when it discards information
12. At one extreme (minimal space cost), the memo table is cleared after each function call
13. At the other extreme (maximal space cost), the memo table persists indefinitely across function calls
14. How long does the completed computation persist in memory and in what form?
15. Gordon Logan’s “instance theory of automaticity” is a general theory of cognitive skills acquisition
16. Each time an algorithm is called, its output is stored in memory
17. There’s a race between algorithm and memory
18. Behavioral outputs are based on the first process to complete
19. Reuse operates “top-down” by storing outputs of completed computations in a memo-table (long term declarative memory)
20. The need to go beyond the reuse of individual instances
21. Studies of motion perception: we represent multipart objects using a hierarchy of relative coordinate frames
22. The unit of computational reuse is not whole instances
23. Humans learn representations that permit effective partial reuse and generalization
24. The prior captures knowledge about the frequency of occurrence for a hypothesis
25. Agents using the Markov property to plan and infer

26. Kalman filter is an example of cached inference
27. It implements exact inference in a linear Gaussian hidden Markov model
28. It resembles non-probabilistic error-driven learning rules
29. It has been proposed as a psychologically plausible model of learning
30. It stores intermediate results (the posterior sufficient statistics) in a cache…
31. Updating them in closed form at the next time step
32. Keeping the computational cost of inference bounded even when dealing with an infinite stream of data
33. Variational Autoencoder: a neural network that maps data inputs to an approximate posterior…
34. The neural network functions as a distributed memo table…
35. An example of a more general family of algorithms known as amortized inference…
36. Which replaces iterative computation with a fast parametrized "recognition model"
37. A "recognition network" is a recognition model that takes the form of a neural network parameterized by synaptic weights…
38. They require a lot of data and struggle to generalize compositionally…
39. They are very expressive and can interpolate well even in complex, high dimensional inference problems…
40. Such as in NLP and visual recognition
41. A key implication of amortized inference: past inferences should exert an influence on future inferences
42. Past inferences imprint themselves on the parameters of the recognition model
43. Given a capacity-limited neural network, amortization using a neural network leads to sharing of structure between different posteriors
44. People are much more accurate at Bayesian inference if the probabilities are realistic
45. The "preferential allocation" hypothesis: the recognition network preferentially allocates its limited capacity to answering high probabilities queries
46. Exact inference and exact planning
47. The precise mechanism of reuse in efficient planning is unknown
48. The Markov property: the rewards and transitions depend only on the current state and action
49. Generating fictive experience offline
50. Making decisions based on stale information
51. Partial cache updating… must prioritize some updates over others
52. When more cognitive resources, time, or incentives are available, humans have a greater propensity for cache updating
53. Entire sequences of actions are remembered as a unit and reused later when an agent occupies the same starting state.
54. In the low-data regime… episodic memories may be the agent's best bet.
55. How does intelligent behavior arise despite limited computational power?

56. Temporal difference learning…
57. Approximates value iteration when the Bellman equation cannot be computed…
58. The expectation over states is unavailable or intractable
59. Samples from the environment and incrementally updates value estimates based on each sample…
60. An example of a “model-free” reinforcement learning algorithm